Color and Shape Recognition

Smruti Saurav Shasani
(111CS0607)

Ramiya Ranjan Meher
(111CS0137)

Manoj Kumar Patra
(111CS0465)

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

Color and Shape Recognition

Thesis submitted in partial fulfillment
of the requirements for the degree of

Bachelor Of Technology

in

Computer Science and Engineering

by

Smruti Saurav Shasani
(111CS0607)

Ramiya Ranjan Meher
(111CS0137)

Manoj Kumar Patra
(111CS0465)

under the supervision of

Dr. Pankaj Kumar Sa

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, Odisha, India

May 2015

Department of Computer Science and Engineering

National Institute of Technology, Rourkela

Certificate

This is to certify that the work in the thesis entitled ***Color and Shape Recognition*** by ***Smruti Saurav Shasani (111CS0607), Ramiya Ranjan Meher (111CS0137), and Manoj Kumar Patra (111CS0465)***, is a record of an original research work carried out by them under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

Date: May 11, 2015

Dr. Pankaj Kumar Sa
Assistant Professor
CSE Department
NIT Rourkela

Acknowledgement

This thesis has been possible due to the help and endeavor of many people. Many individuals have helped throughout for the finishing of this task and each of their inputs has been valuable. Our deepest and sincerest gratitude goes to our supervisor, Dr. Pankaj Kumar Sa, Assistant Professor, Department of Computer Science & Engineering, for his guidance and encouragement throughout the period of the undertaking. He stimulated us to work on the topic and provided valuable information which helped in completing the project through various stages.

We are also thankful to NIT Rourkela for the academic resources it has supplied us through the course of the project. We want to thank the staff from the office who have been sufficiently benevolent to extend their help towards the completion of this project.

We would also like to take this opportunity to express our sincere gratitude to everyone who has provided us with inspirational words, ideas, constructive criticism, and above all, their invaluable time.

Abstract

The object "car" and "cat" can be easily distinguished by humans, but how these labels are assigned? Grouping these images is easy for a person into different categories, but its very tedious for a computer. Hence, an object recognition system finds objects in the real world from an image. Object recognition algorithms rely on matching, learning or pattern recognition algorithms using appearance-based or feature-based techniques.

In this thesis, the use of color and shape attributes as an explicit color and shape representation respectively for object detection is proposed. Color attributes are dense, computationally effective, and when joined with old-fashioned shape features provide pleasing results for object detection. The procedure of shape detection is actually a natural extension of the job of edge detection at the pixel level to the difficulty of global contour detection. A tool for a systematic analysis of edge based shape detection is provided by this filtering scheme. This enables us to find distinctions between objects based on color and shape.

Keywords: Color Models, Edge-based Shape Detection

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A solitary image conveys a ton of information in a brief while in light of the fact that we see an image at the same time, though reading or listening to frequently takes essentially more to process the same information. Colors and shapes work in amicability with one another to impart. Accordingly, a comprehension of shapes is key to comprehension the power of color.

Shape detection obliges pre-programming in a mathematical depiction database of the shapes to detect. For instance, assume composing a program which can recognize a triangular shape, a square shape, and a circular shape. We can do it this way: Run contour identification for discovering the boundary line of every shape considered. Then the number of continuous edges is counted. A sharp deviation in line detection implies an alternate line does this by determining the average vector between adjoining pixels. In the event that three lines identified, then it's a triangle. In the event that four lines, then a square. In the event that one closed line, then it is a circle. You can focus more information by measuring angles between lines (rhombus, square, equilateral triangle, etc.)

Complex shapes obliges pattern recognition that is probability analysis. For instance, assume our algorithm expected to perceive between 10 distinct natural products (just by shape), for example, an orange, a bunch of grapes, an apple, a cherry, and so on. How will this be achieved? Since, all of these are circular, however none of them is seamlessly circular. Also, neither all fruits appear to be identical. By utilizing probability, an investigation can be run that says 'gracias, this natural product matches 85% of the attributes of a cherry, yet just 65% the qualities of an apple, so it's more probable a cherry. It is the computational form of an 'educated guess.' We can likewise say if some specific characteristic is available, then it has a 30% higher likelihood of being an orange. The characteristic can be a stem, for example, spikes like on a pineapple, or fuzziness like on a coconut, and so on. This strategy is known as feature detection.

1.1 Computer Vision

It is the study of machines that have the capacity to extract information from an image that is important to tackle some assignment. As a scientific discipline, computer vision is concerned with the hypothesis behind artificial systems that extract information from images. The image information can take numerous structures, for example, video sequences, sees from different cams, or multi-dimensional information from a medicinal scanner. As a technological discipline, computer vision tries to apply its speculations and models to the development of computer vision frameworks. Illustrations of uses of computer vision include system for:

1. Controlling processes (e.g., an industrial robot or an autonomous vehicle).
2. Detecting events (e.g., for visual surveillance or people counting).
3. Organizing information (e.g., for indexing databases of images and mage sequences).
4. Modelling objects or environments (e.g., industrial inspection, medical image analysis or topographical modelling).
5. Interaction (e.g., as the input to a device for computer-human interaction).

1.1.1 OpenCV (the tool)

The OpenCV Library is principally gone for real-time computer vision. Some sample zones would be Human-Computer Interaction (HCI);

Segmentation, Object detection, and Recognition; Gesture Recognition; Face Recognition; Motion Tracking, Ego Motion, and Motion Understanding; SFM (Structure from Motion); and Mobile Robotics.

The OpenCV Library is a gathering of low-overhead, elite operations performed on images. The OpenCV actualizes a wide mixture of tools for image interpretation. It is good with IPL (Intel Image Processing Library) that actualizes low-level operations on computerized images. Notwithstanding primitives, for example, filtering, banalization, pyramids, image statistics, OpenCV is basically an high-level library implementing algorithms for Camera Calibration (calibration techniques), Optical Flow (tracking), and feature (Feature detection), motion analysis (Motion Templates, Estimators), Geometry, Contour Processing (shape analysis), View Morphing (3D reconstruction), object segmentation and recognition (Embedded Hidden Markov Models, Histogram, Eigen Objects).

1.2 Motivation

There is a clear distinction between the two fruits "apple" and "banana", the identification of this distinction for a human is easy, but how can a computer find the difference between those two objects. Identification of an object from its two dimensional image brought up the real interest for this project. This is a paramount topic in the field of computer vision.

1.3 Objective

In this project object recognition has been performed considering color and shape as its prime entities. These two attributes are the central focus for any object recognition system. Different color models have been used along with different data mining algorithms to predict its proper color. The shape recognition system is to be intermingled with color recognition to produce an efficient technique for object recognition.

1.4 Thesis Organization

The thesis is organised as follows:

Chapter 2: Color Recognition
We discuss about the different types of color model.

Chapter 3: Edge-based Shape Detection
We discuss about the various shape detection techniques.

Chapter 4: Simulation and Results.
Presents the results of various color and shape recognition techniques.

Chapter 5: Conclusion and Future Works.
Presents the concluding remarks, with scope for further research works.

Chapter 2

Color Recognition

Colors can be measured in different ways; undoubtedly, a human's impression of colors is a subjective procedure whereby the mind reacts to the jolts that are created when approaching light responds with the few sorts of Cone cells in the eye. Basically, distinctive individuals see the same enlightened object or light source in diverse ways.

2.1 Color Models

A color model is a system for creating a full range of color from a small set of primary colors. A color model is additionally called as color system or color space. The reason for color model is to encourage the determination of color in some standard, by and large acknowledged way.

2.1.1 RGB Color Model

RGB refers to Red, Green and Blue color model. This color model was first portrayed by Thomas Young and Herman Helmholtz in the Theory of Trichromatic Color vision and by using James Maxwell's color triangle in early 19^{th} century.

In RGB color model, every color shows up in its primary spectral segments of red, green and blue. The model is taking into account Cartesian coordinate system. It is an additive color model, on the grounds that red, green and blue lights are included in different blend to recreate a wide range of color.

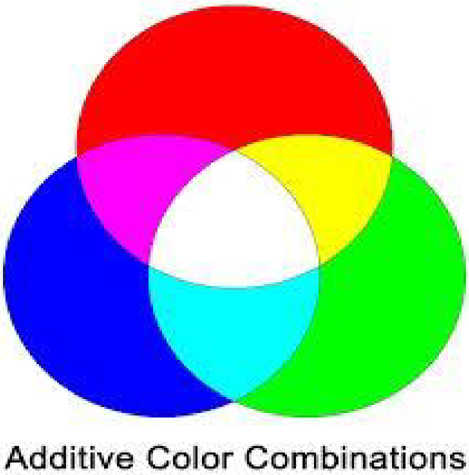

Figure 2.1: RGB Color Wheel.

While representing this color model, a cube is defined on the axes of R, G and B. White color is delivered when each of the primary colors are present at M, where M shows the maximum intensity of the light, let M = 255. The primary diagonal axis uniting the black and white corners defines the intensity.

$$I(R, G, B) = R + G + B$$

All points in a plane perpendicular to the grey axis of the color cube have the same intensity. The plane through the color cube at points $R = G = B = M$ is one such plane. This plane form an equilateral triangle which is the standard RGB chromaticity triangle.

In that cube primary colors (Red, Green and Blue) are at three corners i.e. red in x axis, green in y axis, and blue in z axis, Secondary colors Cyan, Magenta and Yellow are in other three corner i.e. cyan is at diagonally opposite corner of red, magenta is at diagonally opposite corner of green, yellow is at diagonally opposite corner of blue. Black is at origin and White is at corner most remote from origin. Distinctive colors in this model are focuses on or inside the shape, and characterize by vectors stretching out from the origin.

The principle motivation behind the RGB color model is detection, representation, as well as presentation of images in electronic devices,

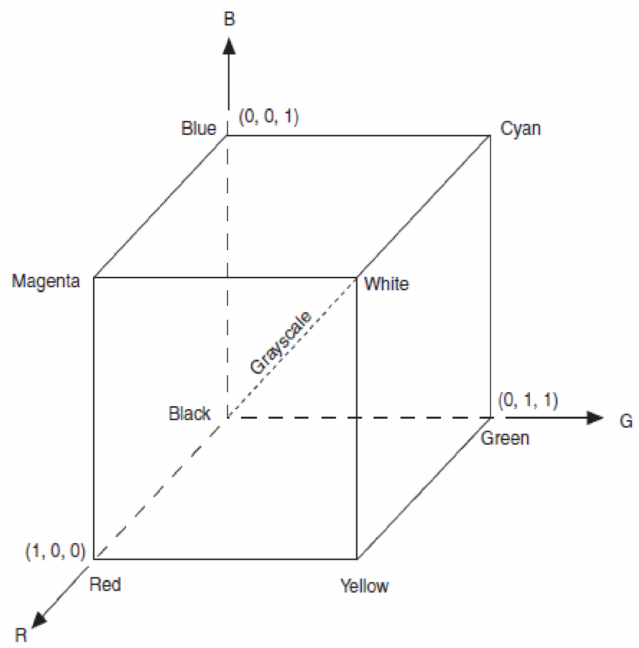

Figure 2.2: Schematic of RGB Color Cube.

for example, TVs, PCs, etc. However it is additionally being utilized as a part of traditional photography. Previously, in the electronic era, this color model as of now had a strong hypothesis behind it, situated in human view of colors.

Numeric Representations

In RGB color model, color is often depicted by specifying the quantity of each of red, green, and blue being incorporated. An RGB triplet (R, G, B) is used to express the color, every segment of which can shift from definite minimum value (usually taken as zero) to a definite maximum value. On the off chance that all the parts are at zero the outcome is dark; if all segments are at highest value, white comes as output.

These extents can be evaluated in several diverse ways. Such as, with any fractional value in between 0 and 1. This depiction is utilized in theoretical analysis, and in arrangements using floating point depictions. Also, every color segment value may also be used as a percentage value, ranging between 0% and 100%. While the component values are frequently used as integers ranging 0 to 255 especially in computers, the range that a 'single 8 bit' byte can offer. Decimal or hexadecimal numbers are often used to represent this. For instance, different RGB notations for brightest saturated red can be denoted as:

Notation	RGB Triplet
Arithmetic	(1.0, 0.0, 0.0)
Percentage	(100%, 0%, 0%)
Digital 8-bit per channel	(255, 0, 0) or sometimes #FF0000 (hexadecimal)

Table 2.1: RGB Numeric Representations

2.1.2 CMYK Color Model

CMY refers to Cyan, Magenta and Yellow. This was invented by Jacob Christoph le Blon (1667 - 1741) that ultimately helped in developing printing systems which includes three-color (CMY) and four-color (CMYK) printing systems. This color model is utilized in various color printing. Presently printing frameworks utilize a system that combines four color including black and some additional color like cyan, magenta, and yellow. This kind of model is called CMYK, where K represents "key" showing black color.

Black is utilized as a part of expansion to CMY for following reasons: Black ink is less expensive as compared to that by blending C, M, and Y to get black. Text imprinted in black has fine detail that would be blurred if printed with three distinct colors. A blemished shade of black is produced by the mixture of C, M, and Y.

The CMYK model works by in part or altogether masking color on a lighter, typically white, background. The ink diminishes the light that would somehow or another be reflected. Such a model is called subtractive because the inks "subtract" brightness from white.

RGB to CMYK Conversion

$$C = 1 - R$$
$$M = 1 - G$$
$$Y = 1 - B$$

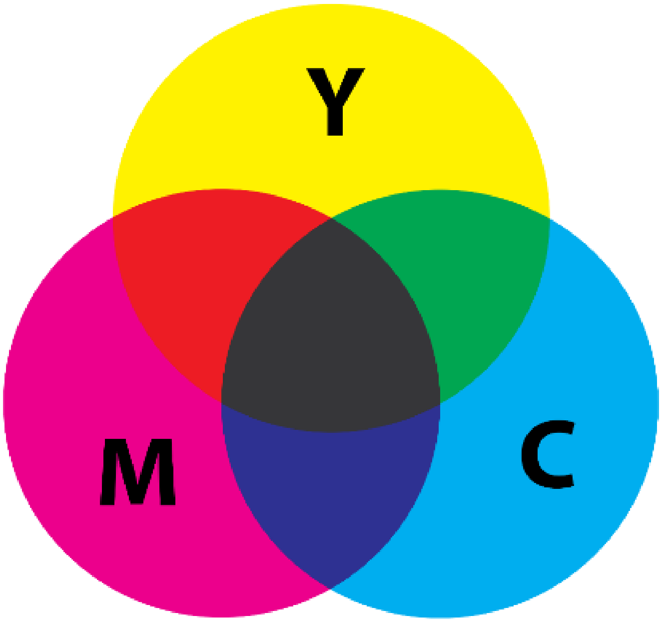

Figure 2.3: CMY Color Wheel.

2.1.3 HSV Color Model

HSV refers to Hue, Saturation and Value. HSV color model was depicted by Alvy Ray Smith in 1978. This model is utilized as a part of computer graphics applications. In HSV, hue represents a saturated color on the outer rim of the Color Wheel. The amount of added white to the color defines Saturation. 0% signifies that the color (at V=100%) is totally white; whereas, 100% signifies totally saturated with no white diluted. Value signifies brightness of the color. 0% indicates pure black or totally dark; 100% speaks full brightness, where hue and saturation determine the actual color.

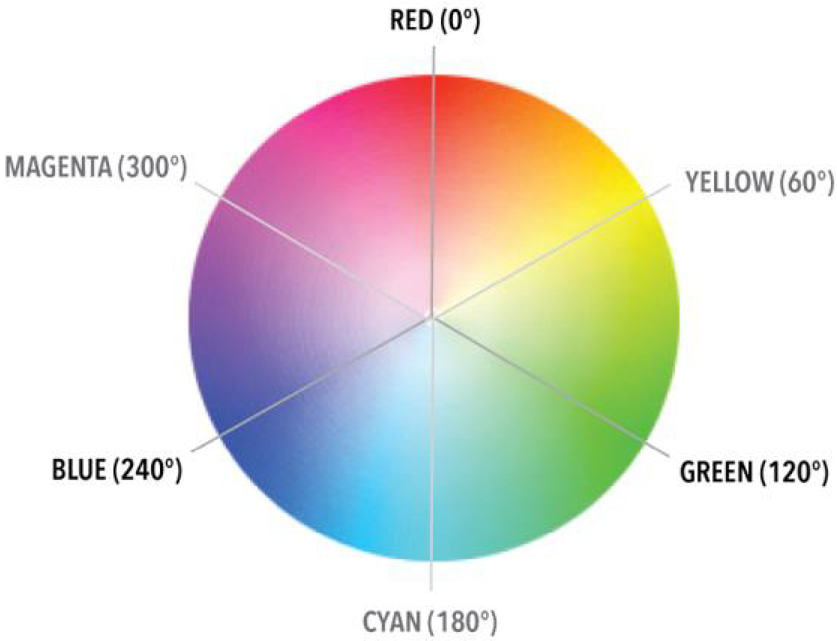

Figure 2.4: HSV Color Wheel.

Color	RGB Values	HSV Values
Red	(255, 0, 0)	(0,100%,100%)
Green	(0, 255, 0)	(120,100%,100%)
Blue	(0, 0, 255)	(240,100%,100%)
White	(255,255,255)	(0,0%,100%)
Black	(0,0,0)	(0,0%,0%)
Cyan	(0,255,255)	(180,100%,100%)
Magenta	(255,0,255)	(300,100%,100%)
Yellow	(255,255,0)	(60,100%,100%)

Table 2.2: RGB and HSV values of some primary colors

2.1.4 Grayscale

A grayscale digital image in photography and computing can be described as an image which contain every pixel as a single-sample, i.e, it holds only information about intensity. Images of this type, also called black and white images, are comprised of different shades of grey, fluctuating from weakest intensity to strongest intensity, i.e., black to white.

Chapter 3

Edge-Based Shape detection

Human vision helps to make usage of a variety of informative sources to detect and recognize an object in an image. Researchers agree to a point that edge and region information are required to obtain a "perceptual unit" in the image at the lowest level of object recognition. Identification of some of the possible invariant features is done and additional signal assets (texture or appearance) are sent to decide whether a point belongs to a particular object or not.

In different circumstances, boundary shape statistics, such as the circular shapes of balls, play a decisive part. Generally local characteristics like the eyes in a human face are quite advantageous. These characteristics aids to provide solid clues for identification, and does not change their behavior in different circumstances. Two-dimensional (2-D) frameworks of these features are adequate enough in various cases, since human beings can identify a scene by drawing an edge.

Perceiving two-dimensional shapes is always an issue in computer vision. Generally "shapes" here signifies Two-dimensional image characteristics of an entity that are actually non-variant to scene factors, or may be their deviation can be effortlessly demonstrated. In the greater part of the enactment, modelling the intensity standards of objects and their background is very difficult. Hence, it is sensible to endeavor the intensity change along the object's margin. The intensity variation along the margin is normally demonstrated as a step edge. The pixels with high intensity gradients are selected first, these pixels ought to be inspected to restrict the off chance that they exist on an estimated shape margin. Different techniques have been implemented to gather local intensity variations into a complete picture depiction.

One of the well-known approach is the generalized Hough transform (GHT). Here the recognized edge pixels choose in favor of a contour in accordance to parametric representation or a table of boundary alignments with relating centroid locations. Since this methodology relies both on the orientations and the positions of edges, it provides reduced localization performance. So, it is likewise difficult to plan the point spread function of the voting process. Presenting a contour using the coefficient vector of a fixed polynomial gets the benefit of effortlessly managing the scale and orientation variations. Then again, this polynomial depiction relies upon the polynomial's order and gives just an inexact contour descriptor. On the account of a polygonal-molded entity, constant polynomial depiction is not effective usually. This type of model crevice is never simple to quantify, creating an efficient mistake investigation troublesome. "Band matching" is applied after contour detection, which is another methodology. An ensemble of static width having the predictable entity shape is placed on the contour image, to find out if the number of edge pixels inside the ensemble is beyond a definite limit, then the shape is supposed to be distinguished. The width of the ensemble provides more vulnerability towards localization, and makes the investigation of the calculating procedure very convoluted.

The edge-based shape recognition systems experience the ill effects of the same issue: information loss during the contour detection phase and the trouble of statistical performance analysis. On inspecting every point so as to decide whether it is a candidate edge pixel or not, then the conclusion is truly established only on local gradient data. Hence, a true boundary pixel, with a noisy gradient estimate, may be eliminated by the threshold applied for detection of edge. Moreover, the locations of identified edges are quite liable to noise. Grouping or fitting stage circulates these errors, and leads to faults in recognition and localization of the shape. Generally the error transmission of contour detection is found to be nonlinear, and the combination or fitting method also seems to be nonlinear in nature; therefore it is not reliable and generally not considered to linearly approximate the propagation of error when the agitation is high enough.

One shortcoming of this methodology is the filtering with numerous geometrical parameters needs enormous amount of computation which can be eased by utilizing multi resolution processing as per complexity of the shape.

3.1 Edge Detection

With a specific end goal to perform a content-based analysis of an image, it is important to find meaningful and significant features from the accumulation of pixels that constitute the image. Contours, lines, blobs, and so on, are fundamental image elements that define an image's content.

We can likewise say that sudden changes of discontinuities in an image are called as edges. Significant transitions within an image are called as edges. The greater part of the shape information of an image is enclosed in edges. So first we identify these edges in an image and by utilizing these filters and after that by enhancing those areas of image which contains edges, sharpness of the image will increase and image will get to be clearer.

Important visual information are carried by edges since they delineate the image elements. For this reason, they can be used, for example, in object recognition. However, simple binary edge maps suffer from two main drawbacks. First, the edges detected are unnecessarily thick. This means precise localization of an object limit cannot be done. Second, and more importantly, it is difficult to find a threshold that is sufficiently low to detect all important image edges of an image and that is, at the same time, sufficiently high to not include too many insignificant edges. This is a trade-off problem that the Sobel algorithm tries to solve.

3.1.1 Sobel Filter

Computer vision and Image processing, widely uses sobel filter, mostly within edge detection algorithms. It produces an image in which mostly edges and transitions are highlighted. It is named after the mathematician Irwin Sobel, who also presented the idea of ”Isotropic 3x3 Image Gradient Operator”.

It is said to be a bidirectional filter because it only affects the vertical or horizontal image frequently depending on which kernel of the filter is used. The Sobel operator is in view of convolving the image with a little, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive as far as computations.

So, the Sobel operator gives an approximation of the image gradient by differencing pixels in the horizontal and vertical directions. Other gradient operators also exists like, Prewitt Operator, Roberts Operator, Laplacian Operator.

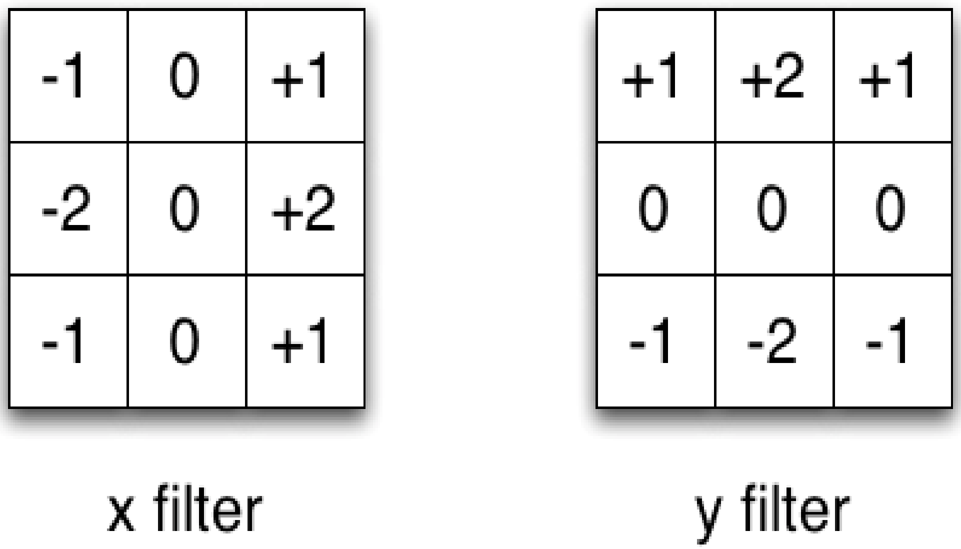

Figure 3.1: Sobel Filter.

3.2 Line Detection

Information regarding the lines or edges in an image is valuable in numerous enactments, especially in Computer Vision tasks. If we know the equations of the same line from several two-dimensional images, then it is conceivable to get the coordinates of the object offering ascent to the two-dimensional images. Extracting the line information physically from an image can be extremely laborious and requires more time particularly if numerous number of lines are present in the image. An involuntary technique is often desirable, however is not as minor as edge detection as one will have to decide which edge point fits to which line. Therefore, this separation possible is possible by using the Hough Transform and is the perfect strategy to be used in our program for involuntary line detection.

3.2.1 Hough Transform

Paul Hough patented the technique of Hough Transform in 1962, is an effective worldwide strategy for edge detection tasks. It changes between the Cartesian space and the Parameter space due to which a straight line or any other boundary information could be characterized.

Across a straight line, the slope of the line always remains constant. So this forms the basics of identifying an edge to be a straight line. We consider a small error factor in the slope so as also to consider those pixels which do not have exactly the same slope value but also their value does not differ much.

3.2.2 Advantages and Disadvantages

The major upside of Hough Transform is that the pixels present on one line need not be adjacent to one another which can be exceptionally valuable when attempting to perceive lines with tiny breaks in them due to noise, or identification in partially occluded images.

With respect to the hindrances of Hough Transform, then it may give ambiguous results when objects are associated by chance. This plainly indicates another drawback that is the recognized lines are infinite lines defined by their (m,c) values, as opposed to finite lines with definite termination points.

3.3 Regular Polygon Detection

Features frequently appear as regular polygons in scenes encompassing manufactured articles. Triangular, rectangular and octagonal signs show acute information in road scenes. Polygonal figures appear regularly in the arrangement of office environments. However, corners are generally partial polygons, however over a substantial array of scales regular polygons indicate far more indoor structure. Also, in outdoor scenes, brickwork, office buildings, windows, show traits that can be identified repeatedly using the technique of regular polygon detection.

Unlike image processing, where data consist of raster images, this suggested algorithm manages drawings in vector configuration, comprising of various line segments. The task of detecting a polygon from a set of line-segments is divided into four major steps. In the first place the intersection of line-segment is identified. Then, a graph is made instigated by the drawing, where proper intersection points of line-segments (or endpoints) are represented by vertices and most relatively open sub-segments that contain no vertices are represented by edges. The next step discovers the Minimum Cycle Basis (MCB) of the graph that is induced in earlier step. In the final step, an arrangement of polygons based on cycles in the earlier found MCB is built. This is straight forward if we convert every cycle into a polygon, where every vertex and every edge in the cycle represents a vertex and an edge in the polygon respectively.

3.4 Circle Detection

The representation of circle in parameter space is

$$r^2 = (x - a)^2 + (y - b)^2$$

A circle is specified with three parameters: the X and Y coordinates of its centre (a & b), and r, its radius.

3.4.1 Circular Hough Transform

Feature extraction techniques, like, Circle Hough Transform (CHT) is used for detecting circles. It is a part of Hough Transform. The reason for this strategy is discovering circles in blemished image. In the Hough parameter space the circle candidates are created by "voting" and afterward nearby maxima is selected in a so-called accumulator matrix.

A change of a point in the x-y plane to the parameter space may be portrayed as the basic behind the General Hough Transform. The parameter space is characterized by space of the object of interest.

The line is difficult to represent in parameter space, contrasted with the circle, transformation of the parameter of the circle to the parameter space can be done easily. The equation of the circle is:

$$r^2 = (x - a)^2 + (y - b)^2$$

where, a,b, and r, are three parameters of circle, where (a,b) is centre of the circle in the direction (x,y) respectively and r is its radius.

Circle can be represented in parametric form as:

$$x = a + r\cos\theta$$

$$y = b + r\sin\theta$$

Along these lines the circle's parameter space shall fit in with three dimensional, though the line just had a place with two dimensional. As the quantity of parameter expected to portray the shape increment and in addition the measurement of the parameter space expand, simultaneously the Hough transform complexity too increases. There for straightforward shape with parameter fitting in with two dimensional or at most three dimensional. The parametric representation of the circle, the range can be held constant or a fixed number of radii can be set.

At every edge-point we draw a circle with centre in the point with the craved range. The drawn circle attracts the parameter space, such that our x-axis, y-axis, and z-axis are a-value, b-value, and the radii respectively. Accumulator matrix has identical size as parameter space. Value in our accumulator matrix is incremented. Hence, we clear over vitality edge-point in the information image drawing circle with the wanted circle with preferred radii increasing and updating the value in our accumulator. Now, when each edge-point and each preferred radius is utilized, then we can get the number of circles passing through the individual coordinate stored at the accumulator. Hence the highest number relate to the centre of the circle in the image.

Chapter 4

Simulation and Results

4.1 Color Recognition

The color recognition algorithm is implemented using both RGB as well as HSV color models.

Figure 4.1: Test Image I for COLOR Recognition

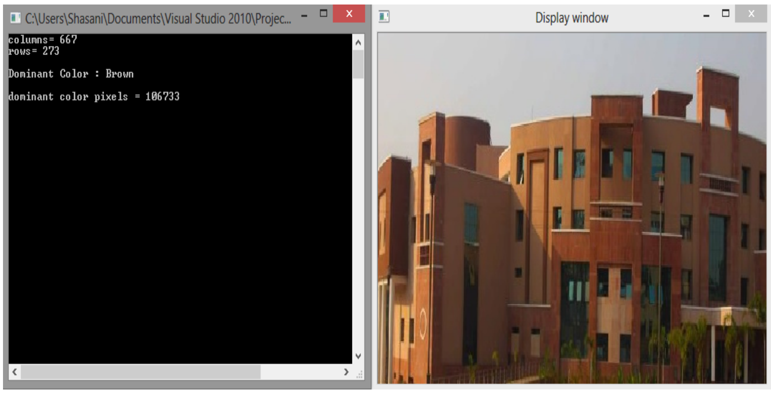

Figure 4.2: Test Image II for COLOR Recognition

4.2 Shape Recognition

4.2.1 Polygon Recognition

Input :

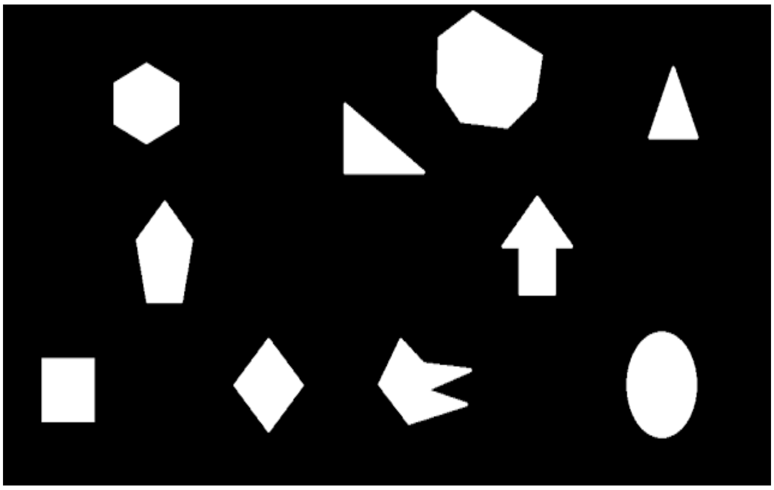

Figure 4.3: Input Image for POLYGON Detection

Output :

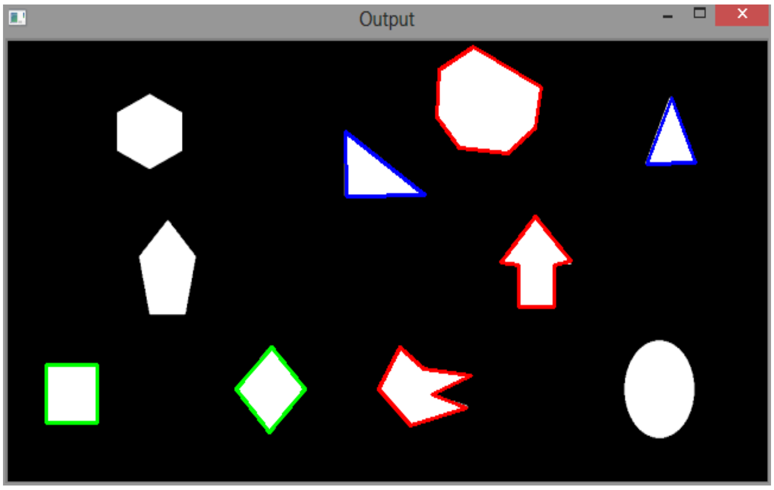

Figure 4.4: Output Image I for POLYGON Detection

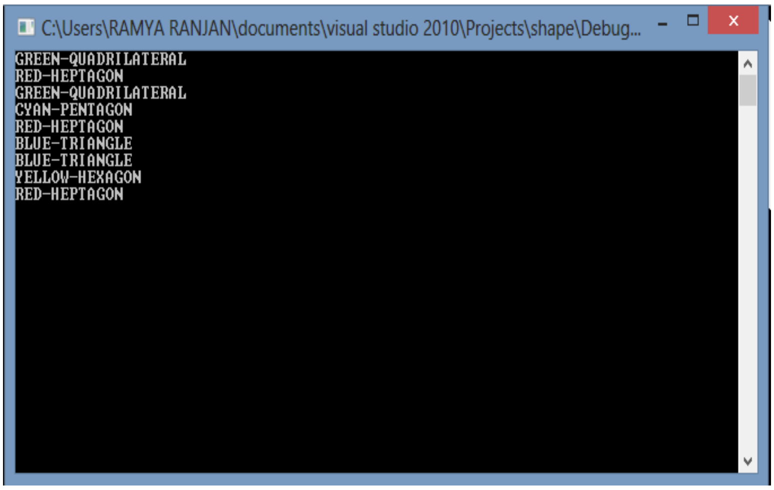

Figure 4.5: Output Image II for POLYGON Detection

4.2.2 Circle Detection

Input:

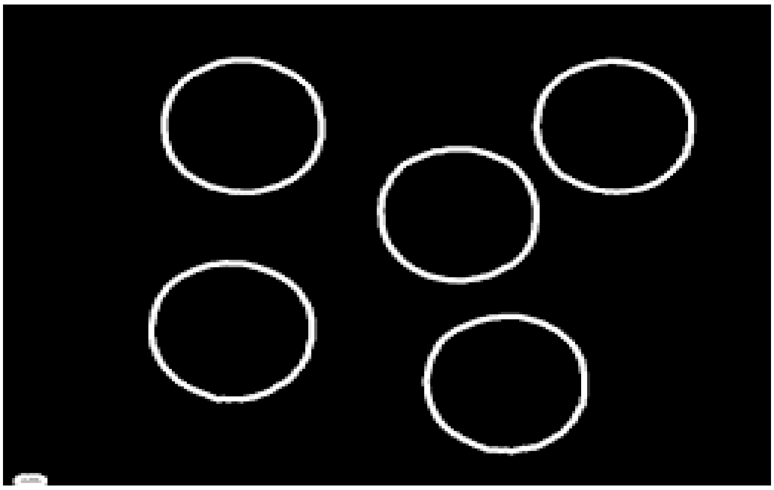

Figure 4.6: Input Image I for CIRCLE Detection

Output:

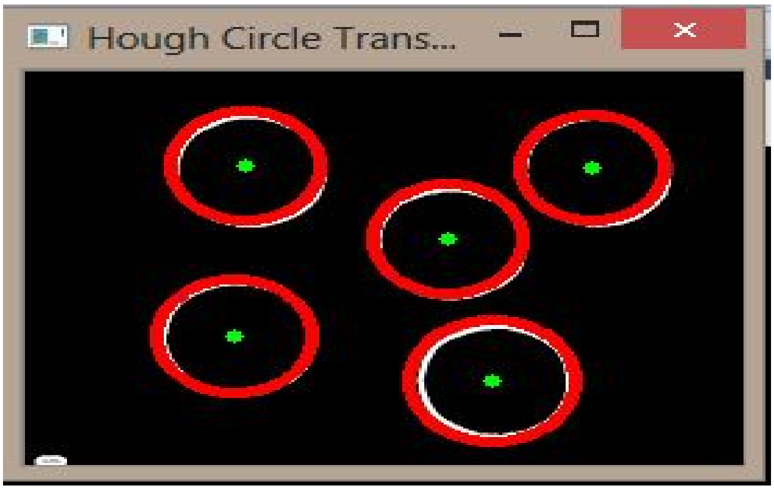

Figure 4.7: Output Image I for CIRCLE Detection

Input:

Figure 4.8: Input Image II for CIRCLE Detection

Output:

Figure 4.9: Output Image II for CIRCLE Detection

Chapter 5

Conclusion and Future Work

We examine the task of integrating color and shape, which forms the base of object detection algorithms. Supreme modern object detectors depend on the shape while overlooking color. Current tactics to augmenting intensity centered detectors with color frequently deliver inferior outcomes for object categories with fluctuating significance of color as well as shape. Our approach uses the color attributes as an unambiguous color representation for object recognition tasks. While, color attributes are dense, computationally efficient, and holds some degree of photometric invariance while keeping discriminative power.

Detection of regular polygons is one of the most significant task with multi-applications in computer vision and robotics. Planar shapes in sketches can be detected using this algorithm. It has been instigated as a working model for shape retrieval and architectural representation from sketches. This algorithm detects all minimal polygons that can be created from a set of line segments in polynomial time and space complexity.

Using the latest, complex and efficient algorithms, there is a chance for improvement in the described algorithm. Many different data mining algorithms have to be used so as to make the proposed project more efficient. Additional work can be performed regarding the detection and correction of rounding errors resulting from finite precision calculations.

Bibliography

[1] Moon, H., Chellappa, R., and Rosenfeld, A., "Optimal Edge-Based Shape Detection," *IEEE Transactions on Image Processing*, Vol.11, No.11, Nov 2002.

[2] Rege Sanket, Memane Rajendra, Phatak Mihir, and Agarwal Parag, "2D Geometric Shape and Color Recognition," *International Journal of Advanced Research in Electrical, Electronics and Instumentation Engineering*, Vol.2, Issue.6, June 2013.

[3] Canny, J.F., "A computational approach to edge detection," *IEEE Trans.Pattern Anal. Machine Intell.*, Vol.8, No.6, pp.679-698, Nov 1986.

[4] Alfredo Ferreira, Manuel J. Fonseca, and Joaquim A. Jorge, "Polygon Detection from a Set of Lines," *Department of Information Systems and Computer Science, INESC-ID/IST/Technical University of Lisbon.*

[5] Cox, I.J., Rehg, J.M., and Hingorani, S.L., "A bayesian multiple hypothesis approach to contour segmentation," *International, Journal of Computer Vision*, Vol.11, pp.05–24, 1993.

[6] Fahad Shahbaz Khan, Rao Muhammad Anwer, Joost van de Weijer, Andrew D. Bagdanov, Maria Vanrell, Antonio M. Lopez, "Color Attributes for Object Detection," *Computer Vision Laboratory, Link oping University, Sweden; Computer Vision Center, CS Dept. Universitat Autonoma de Barcelona, Spain.*

[7] Patel Shalinee, Trivedi Pinal, Gandhi Vrundali, "2D Basic Shape Detection Using Region Properties," *International Journal of Engineering Research & Technology*, Vol.2, No.5, pp.1147-1153, May 2013.

[8] Koschan, A., "A comparative study on color edge detection," *Proceedings of the 2nd Asian Conference on Computer Vision ACCV'95*, pp. 574-578, Singapore, 1995.

[9] Sun, H.X., Zhang, Y.H., and Luo, F.U., "Color edge detection based on HIS color space," *Optical Technique,*, Vol.35, No.11, pp. 221-228, March 2009.

[10] Dutta Soumya, Chaudhuri Bidyut, "A Color Edge Detection Algorithm in RGB Color Space," *International Conference on Advances in Recent Technologies in Communication and Computing*, pp.337-340, 2009.

[11] Laganiere Robert, "OpenCV2 Computer Vision Application Programming Cookbook", PACKT Publishing, 2011.

[12] Bradsky Gary, Kaehler Adrian, "Learning OpenCV," O'Reilly Media Publications, 2008.